Lecture 10

Geometric objects and transformations

Warning: this is a draft copy. It has not been passed any revision

The idea of abstract spaces

- An abstract space is a global or a universe that contains sets
- A set contains abstract objects of certain type
- The space specifies the formal relationships and manipulations (axioms) for
  - Members in each set
  - Members of different sets
- The space formalizes the manipulations and relationships between abstract (abstract means that no details about how the entities are represented or how the manipulation are performed) entities that could serve in certain applications

Example of abstract spaces: Scalars

- Scalars space (also is known as scalar field)
  - Entities: Real numbers or complex numbers, each is called a scalar
  - Manipulation: Any pair of scalars can be combined to form another through addition or multiplication
  - The addition and multiplication stratifies closure associativity, commutability and inverse

Another abstract space: linear or vector space

- Entities: Scalars, points and vectors
  - Scalars are real or complex numbers
  - A point is defined by one property: Its location, it has neither a size nor a shape
  - A vector has a direction and magnitude but has no location
- Manipulation
  - Scalar-scalar manipulation as in scalar space
  - Scalar-vector manipulation (multiplication)
  - Vector-vector manipulation (addition and two types of multiplication: cross and dot)

Another abstract space: Euclidian space

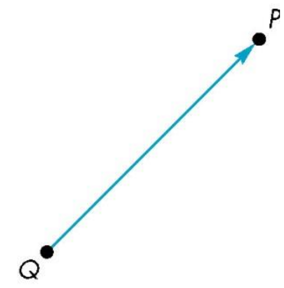
- The same as vector space with the addition of the concept of distance between two points
- Cartesian coordinate system implements the Euclidian spaces by providing how the points and vectors are represented and written and how the cross and dot product are performed in that representation

Geometric objects and computer graphics

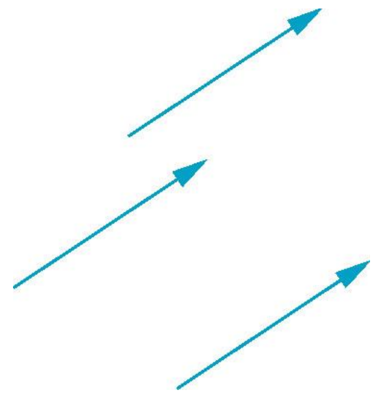
- Objects are represented by geometric entities in a graphics application
- The basic geometric entities and the relationships between them can be described using three fundamental abstract primitives: Scalars, points and vectors
- Which space could support the needs of computer graphics operations (it seems that Euclidean space is suitable, it includes all the primitives)

Line segment and a vector

- A directed line segment has a length (magnitude) and direction. Could we consider it as a Vector? No, because it has a location!
- In most graphics applications, we need to deal with line segment that connects two points (from Q to P, for example). How can we differentiate this from the location less vector that points in the direction from Q to P and has magnitude equals to the distance between Q and P? Euclidian space failed in supporting computer graphics.
- Also, in computer application we often need to specify a point using another point and a vector(staring from Q , required a point P that lies at a distance m from Q in the direction of vector v). Again, Euclidian space fail to support this because it does not contain point-vector manipulations>
- Zero vector is a vector that has a magnitude of zero and undefined direction



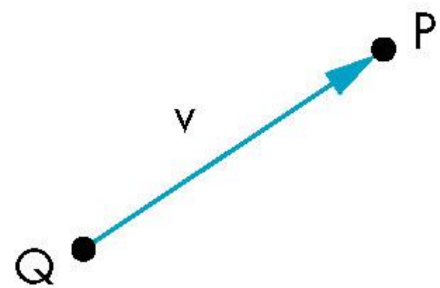
A directed line segment from point Q to point P



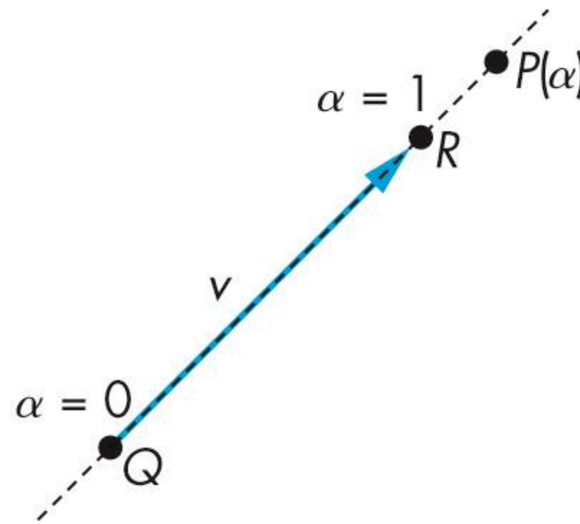
Identical vectors have the same direction and magnitude of the line segment

Affine space

- Affine space is an extension to Euclidian space
- It adds a manipulation that combine a point and a vector to yield a point in what is called point-vector addition
- Point vector addition represent moving from one location to another through a direction (vector) a certain distance (vector magnitude)
- As shown in the last equation, The point vector addition, makes some point-point addition / scalar-point multiplications make sense: The one in which the sum of the two added points coefficients is one . This is also called parametric line segment: Reaching a point starting from Q in the direction of the vector from Q to R using a parameter alpha



$$P = Q + v \qquad v = P - Q$$



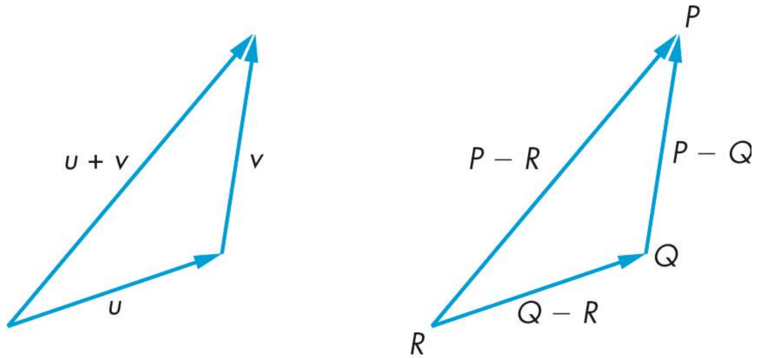
$$P = Q + \propto (R - Q) = \propto R + (1 - \propto) Q$$

Abstract spaces and Abstract Data types(ADT)

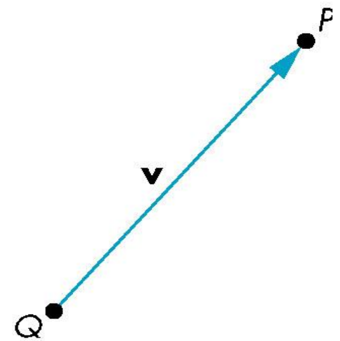
- In computer science, The equivalent to abstract entity is ADT.
- The ADT define the entities and the operation that could be done on that entities regardless how the entity is represented or how the operation are implemented for a specific representation
- For example. The integer as an ADT is an integral number that can be added to/multiplied by to another integer. The number itself could be represented differently on different systems on the binary level, hence the implementation of the operation. But each system has to implements the characteristics of the integer ADT
- Another example, the vector as an ADT allow vector-vector addition and multiplication and also scalar-vector multiplication but the implementation of these operation and the representation of vectors are different if we user polar or Cartesian coordinate systems

Geometric ADT (Affine space)

Dealing with geometric ADT regardless any coordinate system or frame used to represent them



(a) (b)



$$v = P - Q$$

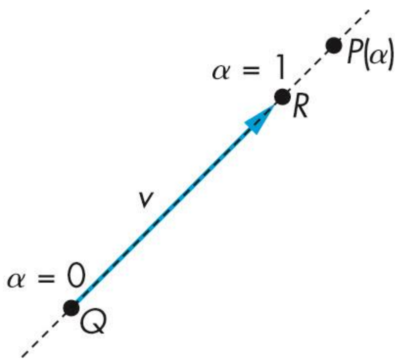
$$u + v$$

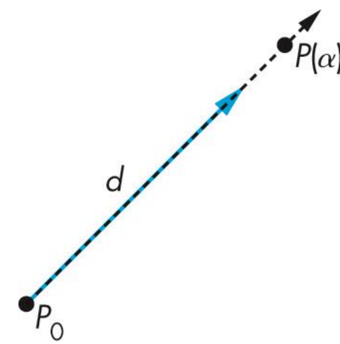
$$(P - Q) + (Q - R) = P - R$$

$$|\propto v| = |\propto||v|$$

Geometric ADT (Affine space)

Dealing with geometric ADT regardless any coordinate system or frame used to represent them





$$P = Q + \propto v$$

$$v = R - Q$$

$$P = Q + \propto (P - Q) = \propto R + (1 - \propto) R$$

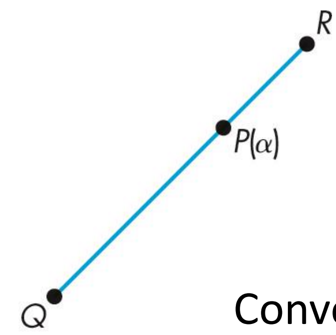
$$P = \propto_1 R + \propto_2 Q$$

Point addition is allowed under the condition that the operation represent a point vector addition because affine space define point vector addition but not point-point addition. The condition is satisfied if the sum of the two points coefficients is one

Geometric ADT (Affine space)

Dealing with geometric ADT regardless any coordinate system or frame used to represent them

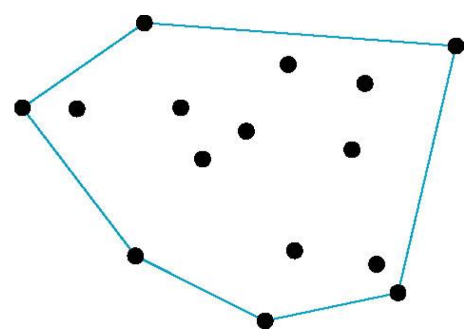
Convexity of a geometric object: A convex object is one for which any point lying on the line segment connecting any two points in the object is also in the object. Convexity is an important property in computer graphics



Convex object

$$P = \alpha_1 P_1 + \alpha_2 P_2 + \ldots + \alpha_n P_n$$

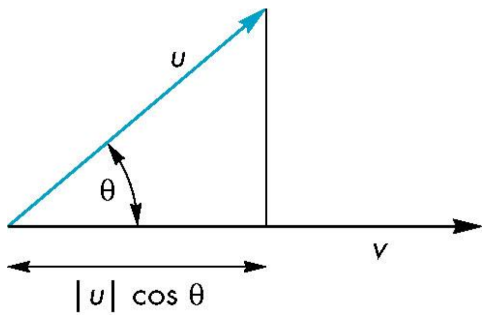
The sum is only defined if the sum of all alpha is one. If each alpha is greater than or equals zero, P give the convex hull of the set of points



Convex hull

Geometric ADT (Affine space)

Dealing with geometric ADT regardless any coordinate system or frame used to represent them

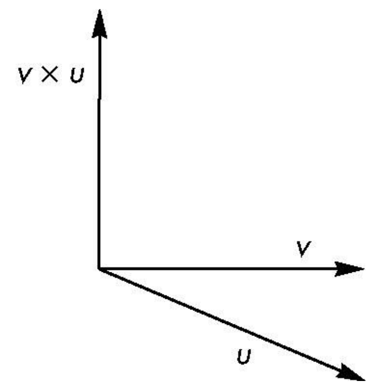


Dot product

$$|u|^2 = u.u$$

$$\cos(\theta) = \frac{u.v}{|u||v|}$$

Cross product: right-hand rule or right screw



$$n = u.u$$

$$|\sin(\theta)| = \frac{|u \times v|}{|u||v|}$$

Geometric ADT (Affine space)

Dealing with geometric ADT regardless any coordinate system or frame used to represent them

Three points not on the same line determine a unique plane

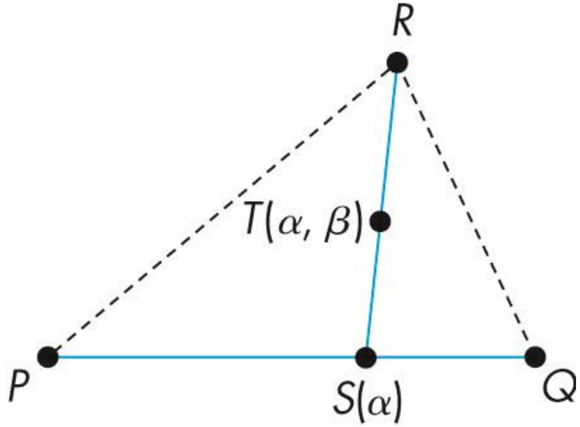
$$S(\propto) = \propto P + (1-\propto)Q, \qquad 0 \leq \propto \leq 1$$

$$T(\beta) = \beta S + (1-\beta)R, \qquad 0 \leq \beta \leq 1$$

$$T(\propto,\beta) = \beta[\propto P + (1-\propto)Q] + (1-\beta)R$$

Rearranging

$$T(\propto,\beta) = P + \beta(1-\propto)(Q-P) + (1-\beta)(R-P)$$



Which shows that a plan can also be uniquely defined using any two non parallel vectors and a point

Geometric primitives and the representation of real world objects in computer graphics application

Real-world objects are three dimensional objects but they have three characteristics that fit them with existing graphics hardware that deals with simple planner geometric objects

- The objects are described by their surfaces and can be though as being hollow
- The objects can be specified through a set of vertices in 3D
- The objects either are composed of or an be approximated by flat, convex polygons

Then we can represents any object in the real world in a graphics application using the points, scalars, and vectors. Now, how we can specify a specific point or a specific vector to the application? We need a reference frame (three linearly independent vectors and an origin) and we need to know how to represent the points and vectors using the reference frame or coordinate system